REMARKS

[0003] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-41 are presently pending. Claims 1, 5, 11, 18, 19, 24, 26, 31, 36 and 41 amended herein. No claims are withdrawn or canceled herein. No new claims are added herein.

Statement of Substance of Interview

[0004] Examiner Anya graciously met with me—the undersigned representative for the Applicant—on November 4th, 2008, at the USPTO. Applicant greatly appreciates the Examiner's willingness to meet. Such willingness is invaluable to both of us in our common goal of an expedited prosecution of this patent application.

[0005] During the interview, I discussed how the claims differed from the cited references, namely Chiang and Roberts. Without conceding the propriety of the rejections and in the interest of expediting prosecution, I also proposed possible clarifying amendments.

[0006] The Examiner was receptive to the proposals, specifically the clarification regarding the programming languages. However, the Examiner indicated that he would need to review the cited references more carefully and do another search, and requested that the proposed amendments be presented in writing.

[0007] Applicant herein amends the claims in the manner discussed during the interview. Accordingly, Applicant submits that the pending claims are allowable over the cited art of record for at least the reasons discussed during the interview.

Formal Request for an Interview

[0008] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0009] Please contact me to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for me, I welcome your call as well. My contact information may be found on the last page of this response.

Claim Amendments

[0010] Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1, 5, 11, 18, 19, 24, 26, 31, 36 and 41 herein. Applicant amends claims to clarify claimed features. Such amendments are made to expedite prosecution and more quickly identify allowable subject matter. Such amendments are merely intended to clarify the claimed features, and should not be construed as further limiting the claimed invention in response to the cited references.

[0011] Support for the amendments to claims 1, 5, 11, 18, 19, 24, 26, 31, 36 and 41 is found in the specification at least at pages 14-16.

Formal Matters

Specification

[0012] The Examiner objects to the specification for not having updated statuses in relation to cross-referenced Patent Applications. Herein, Applicant amends these paragraphs, as shown above, to correct the informalities noted by the Examiner. Applicant notes that patent applications serial numbers 09/902,812 and 09/901,555 are still pending, and thus have not been issued as Patents nor have the applications been abandoned.

Substantive Matters

Claim Rejections under § 112 6TH ¶

[0013] Claims 19-23 are objected to under 35 U.S.C. § 112, 6th ¶. Applicant respectfully traverses this objection. Furthermore, in light of the amendments presented herein, Applicant submits that the objection is moot. Accordingly, Applicant asks the Examiner to withdraw the objection.

Claim Rejections under § 101

[0014] Claims 5-10 are rejected under 35 U.S.C. § 101. Applicant respectfully traverses this rejection. Furthermore, in light of the amendments presented herein, Applicant respectfully submits that these claims comply with the patentability requirements of §101 and that the §101 rejections should be withdrawn. Accordingly, Applicant asks the Examiner to withdraw these rejections.

[0015] If the Examiner maintains the rejection of these claims, then Applicant requests additional guidance as to what is necessary to overcome the rejection.

Claim Rejections under § 103

[0016] Claims 1-41 are rejected under 35 U.S.C. § 103. In light of the amendments presented herein and the discussion during the above-mentioned Examiner interview, Applicant submits that these rejections are moot. Accordingly, Applicant asks the Examiner to withdraw these rejections.

[0017] The Examiner's rejections are based upon the following references in combination:

- **Yach:** *Yach, et al.,* US Patent Publication No. 2002/0112078 (published August 15, 2002);
- **Roberts:** *Roberts, et al.,* US Patent No. 6,792,605 (issued September 14, 2004);
- **Chiang:** *Chiang*, US Patent Publication No. 2006/0294500 (published December 28, 2006); and
- **Firth:** *Firth, et al.,* US Patent No. 5,987,517 (issued November 16, 1999).

Overview of the Application

[0018] The Application describes an application program interface (API) that provides a set of functions for application developers who build Web applications on Microsoft Corporation's .NET™ platform.

Cited References

Yach

[0019] Yach describes browsing documents at a client device while not requiring a traditional document browsing application. In order to achieve browsing without a browsing application, the client device first transmits an information request to a host system. The host system retrieves the requested information from one or more information sources that store the information. A translation component receives the information from the host system and translates the information from a plurality of content types into a virtual machine language program. The virtual machine language program is then transmitted to the client device, which executes the virtual machine language program in order to display and interact with the information.

Roberts

[0020] Roberts describes accessing and using services and applications from a number of sources utilizing a customized application. The present invention accomplishes this through an entity referred to as a web service. The web services architecture maintains a directory of services available to provide processing or services, along with the location of the services and the input/output schemas required by the services. When a request for data or services is received, appropriate services are invoked by a web services engine using service drivers associated with each service. A web services application is then generated from a runtime model and is invoked to satisfy the request, by communicating as necessary with services in proper I/O formats. In one embodiment, the web services application provides responses in the form of HTML that can be used to generate pages to a browser.

Chiang

[0021] Chiang describes providing a web application and generating the basis for a complete web application source code. Based on user interface input files provided by graphic designers, the web application or generates application framework code, an event handler skeleton and s logic foundation code. Web developers then prepare additional source code object-oriented programming language based on the event handler skeleton and business logic foundation code to create web application business logic objects and handler methods. Ultimately the graphical user interface input files prepared by the designers and web application source code prepared by the web developers dynamically bound at runtime.

Firth

[0022] Firth describes a library of reentrant networking functions organized with file system semantics which allow a client application on a client computer connected to a computer network to establish communications with and exchange information with a server application on a server network computer. The library of reentrant networking functions are organized with file system semantics and parallel the function, structure and organization of a file system. Individual reentrant networking functions provide multiple networking features. The reentrant networking functions also provide asynchronous operations and security features. The library of reentrant networking functions can be included in, and called from multiple client applications. This library of reentrant networking function simplifies the creation of client applications such as network browsers that communicate with the Internet or an intranet computer network.

Obviousness Rejections

Lack of *Prima Facie* Case of Obviousness (MPEP § 2142)

[0023] Applicant disagrees with the Examiner's obviousness rejections. Arguments presented herein point to various aspects of the record to demonstrate that all of the criteria set forth for making a prima facie case have not been met.

Based upon Yach and Roberts

[0024] The Examiner rejects claim 1 under 35 U.S.C. § 103(a) as being unpatentable over Yach and Roberts. Applicant respectfully traverses the rejection of this claim and asks the Examiner to withdraw the rejection of this claim. Applicant notes that the Examiner has previously used the Yach reference to address claimed features in this Application, for example, in the Office Action issued May 4th, 2006. In response to this Office Action, Applicant filed an appeal brief on December 1st, 2006, from which prosecution was reopened and subsequent Office Action issued. Thus, Applicant submits that claim 1 is patentable over the purported combination of Yach and Roberts for at least the reasons set forth in the appeal brief filed December 1st, 2006, and Yach should be removed as a reference cited against the claim accordingly.

<u>Based upon Chiang and Roberts</u>

[0025] The Examiner rejects claims 1-8, 10-16, 19-22, 24-29, 31-34 and 36-39 under 35 U.S.C. § 103(a) as being unpatentable over Chiang in view of Roberts. Applicant respectfully traverses the rejection of these claims and asks the Examiner to withdraw the rejection of these claims.

<u>Independent Claim 1</u>

[0026] Applicant submits that the combination of Chiang and Roberts does not teach or suggest at least the following elements as recited in this claim, as amended (with emphasis added):

> "A software architecture implemented at least in part by a computing device for a distributed computing system comprising:
>
> a plurality of applications configured to handle requests submitted by remote devices over a network, wherein the plurality of applications are written in ***different programming languages***;
>
> an application program interface to present functions used by the plurality of applications to access network and computing resources of the distributed computing system; and
>
> ***a common language runtime layer*** that translates the plurality of applications ***written in different programming languages*** into an intermediate language, the intermediate language being:
>
> executed natively by the common language runtime layer; and configured to access resources or services requested by the remote devices, whereby a ***seamless integration between multi-language application***

development is allowed and a robust and secure execution environment for ***multiple programming languages*** is provided."

[0027] The Examiner indicates (Action, p. 5-6) the following with regard to this claim:

11. As to claim 1, Chiang teaches a software architecture implemented at least in part by a computing device for a distributed computing system comprising: a plurality of applications configured to handle requests submitted by remote devices over a network (Web Application 400 page 3 paragraphs 0033, "...input files..." page 3 paragraphs 0035/0036, Input 605 page 4 paragraph 0039); an application program interface to present functions used by the plurality of applications to access network and computing resources of the distributed computing system (Application Framework 410 page 3 paragraph 0033); and a common language runtime layer that translates the plurality of applications written in different languages into an intermediate language (Web

Application Source Code Output 610), the intermediate language being: executed natively by the common runtime layer ("The web application generator 205 generates the application framework code 605...regardless of the language format for the input files, because the input tags are interpreted in the same fashion for the different languages...") and configured to access resources or services requested by the remote devices whereby a seamless and robust integration between multi-language application development is allowed (Web Application Generator 205 page 4 paragraphs 0039 – 0044, page 5 paragraph 0050, page 6 paragraphs 0064, page 7 paragraphs 0068/0069).

Chiang is silent with reference to providing secure execution environment for multiple programming languages.

Roberts teaches providing secure execution environment for multiple programming languages ("...access control..." Col. 4 Ln. 36 – 38, Col. 6 Ln. 1 – 9, Ln. 47 – 63).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Chiang with the teaching of Roberts because the teaching of Roberts would improve the system of Chiang by providing the essential services of *identification and authentication* (I&A), *authorization*, and *accountability* where identification and authentication determine who can log on to a system, and the association of users with the software subjects that they are able to control as a result of logging in; authorization determines what a subject can do and accountability identifies what a subject (or all subjects associated with a user) did.

[0028] In this Action, the Examiner equates programming languages, as recited in claim 1, to input files that can be formatted in any "mark-up language" as taught in Chiang Para [0038-0040]. However, Applicant submits the multiple mark-up languages as recited in Chiang are not equivalent to the "programming languages" as recited in claim 1.

[0029] Differences between "mark-up languages" such as HTML and XML, and "programming languages" such as C++, Jscript and Perl are discussed in the application as originally filed consistent with the meaning set forth in <u>Computer Dictionary</u> Fifth Edition, Copyright 2002, Microsoft Corp. pp. 328, 329, and 426, (hereinafter "Exhibit A"). Exhibit A, attached to this response, defines a programming language as "any artificial language that can be used to define a sequence of instructions that can ultimately be processed and executed by the computer," (page 426, Exhibit A). In contrast, a mark-up language is defined as "a set of codes in a test file that instructs a computer how to format the file on a printer or video display or how to index and link its contents," (pages 328 and 329, Exhibit A). Thus, a mark-up language (as defined) is not equivalent to a programming language. Furthermore, differences between a programming language and a mark-up language were well known to one of ordinary skill in the art at the time of the invention.

[0030] As seen in Fig 6 of Chiang and discussed in Para [0039-0040], a web application generator reads a set of web application screens as the input and generates web application source code as the output. The input files include tags corresponding to a mark-up language, and the same output code is generated regardless of the format of the input files, because the input tags are interpreted in the same fashion for the different languages (e.g. XML, HTML, etc.)

[0031] However, Chiang does not teach or suggest "a common language runtime layer that translates the plurality of applications written in different *programming* languages into an intermediate language, the intermediate language being: executed natively by the common language runtime layer" or "*multi-language* application development" as recited in claim 1.

[0032] In contrast, Chiang teaches that computer programmers use only a single programming language, such as the Java language, to write the underlying function of the web application, (Chiang Para [0026]). Furthermore, the web application source code output is generated as a corresponding Java class, (Chiang Para [0049]). This teaching does not address the claimed element specifying multi-language application development and an intermediate language being executed natively by the common runtime layer as specified in claim 1.

[0033] Roberts teaches accessing and using services and applications from a number of sources via a centralized location, i.e. web services architecture. The web service architecture provides access, control, organization, interface, definition, management and operation of a set of published web services, (Roberts, Col 4 Lines 30-47).

[0034] Roberts does not teach or suggest different programming languages, a common language runtime layer, an intermediate language or multi-language application development as specified in claim 1, and therefore Roberts does not account for the deficiencies in Chiang as explained above.

[0035] As shown above, the combination of Chiang and Roberts does not teach or suggest all of the elements and features of this claim. Accordingly, Applicant asks the Examiner to withdraw the rejection of this claim.

Independent Claims 5, 11, 19, 24, 26, 31 and 36

[0036] Similarly, independent claims 5, 11, 19, 24, 26, 31 and 36 each include at least one feature similar to the claimed features not taught or suggested by the combination of Chiang and Roberts as explained above with respect to claim 1. Thus independent claims 5, 11, 19, 24, 26, 31 and 36 are allowable over the cited references for at least similar reasons as claim 1. Accordingly, Applicant asks the Examiner to withdraw the rejection of these claims.

Dependent Claims 2-4, 6-8, 10, 12-16, 20-22, 25, 27-29, 32-34 and 37-39

[0037] These claims ultimately depend upon one of independent claims 1, 5, 11, 19, 24, 26, 31 or 36. As discussed above, claims 1, 5, 11, 19, 24, 26, 31 and 36 are allowable over the cited references. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable over the cited references for at least the same reasons that the base claim is allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Based upon Chiang, Roberts and Firth

[0038] The Examiner rejects claims 9, 17-18, 23, 30, 35 and 40-41 under 35 U.S.C. § 103(a) as being unpatentable over Chiang in view of Roberts, and further in view of Firth.

[0039] Independent claims 18 and 41 include at least one claimed element explained above with respect to claim 1. Applicant submits that Firth fails to account for the deficiencies discussed above in the combination of Chiang and Roberts. As a result, independent claims 18 and 41 are allowable over the cited references for at least similar reasons as those explained above with respect to claim 1.

[0040] Furthermore, claim 41 has been amended to recite "the plurality of programming languages comprising: Visual Basic; C++; C#; COBOL; Jscript; Perl; Eiffel; and Python." As previously explained, the combination of Chiang, Roberts and Firth does not teach or suggest "different programming languages" as recited in claim 41. Therefore, the references alone or in combination, to not teach or suggest the specific programming languages as recited in claim 41.

Dependent Claims 9, 17, 23, 30, 35 and 40

[0041] These claims ultimately depend upon one of independent claims 5, 11, 19, 26, 31 or 36. As discussed above, claims 5, 11, 19, 26, 31 and 36 are allowable over the cited references. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable is also allowable over the cited references for at least the same reasons that the base claim is allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Dependent Claims

[0042] In addition to its own merits, each dependent claim is allowable for the same reasons that its base claim is allowable. Applicant requests that the Examiner withdraw the rejection of each dependent claim where its base claim is allowable.

Conclusion

[0043] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action**. Please call or email me at your convenience.

Respectfully Submitted,

Lee & Hayes, PLLC
Representatives for Applicant

__/Jacob Rohwer 61,229/______________ Dated: _11/10/2008____
Jacob P. Rohwer (jacob@leehayes.com; 509-868-8323)
Registration No. 61,229
Bea Koempel-Thomas (bea@leehayes.com; 509-944-4759)
Registration No. 58,213
Customer No. **22801**

Facsimile: (509) 323-8979
www.leehayes.com